

# Regressão logística e árvores de regressão e classificação

## Aula 4

Magno Severino  
PADS - Modelos Preditivos  
22/05/2021

# Nas aulas passadas...

- Erro de treino e erro de predição.
- Decomposição do erro em viés e variância.
- Estimação do erro de predição: separação em treino-teste, validação cruzada.
- **Seleção de subconjuntos:** considera um subconjunto das  $p$  preditoras:
  - Best subset selection;
  - Forward stepwise selection;
  - Backward stepwise selection.
- **Regularização:** ajusta-se um modelo com as  $p$  preditoras e os coeficientes estimados são encolhidos em direção a zero:
  - Ridge;
  - LASSO;
  - Elastic-net.

# Objetivos de aprendizagem

Ao final dessa aula você deverá ser capaz de

- definir um problema de classificação;
- conceituar o modelo KNN para classificação;
- conceituar a regressão logística;
- conceituar uma árvore de classificação/regressão;
- avaliar performance de um modelo de classificação.





# Problemas de classificação

# Problemas de classificação

- Situações em que o objetivo é assinalar uma classe à uma observação.
- Dados `Default`<sup>1</sup>:
  - Informações sobre 1000 clientes;
  - **default**: indica se o cliente apresentou *default*;
  - **student**: indica se o cliente é estudante;
  - **balance**: saldo médio mensal no cartão de crédito;
  - **income**: renda do cliente;
- Objetivo: prever quais clientes apresentarão *default* no cartão de crédito.

[1] Fonte: dados no pacote `ISLR`, do livro *An Introduction to Statistical Learning with Applications in R*.

# Dados Default

	default 	student 	balance 	income 
1	No	No	729.53	44361.63
2	No	Yes	817.18	12106.13
3	No	No	1073.55	31767.14
4	No	No	529.25	35704.49
5	No	No	785.66	38463.5
6	No	Yes	919.59	7491.56
7	No	No	825.51	24905.23
8	No	Yes	808.67	17600.45
9	No	No	1161.06	37468.53
10	No	No	0	29275.27

Showing 1 to 10 of 10,000 entries

Previous

1

2

3

4

5

...

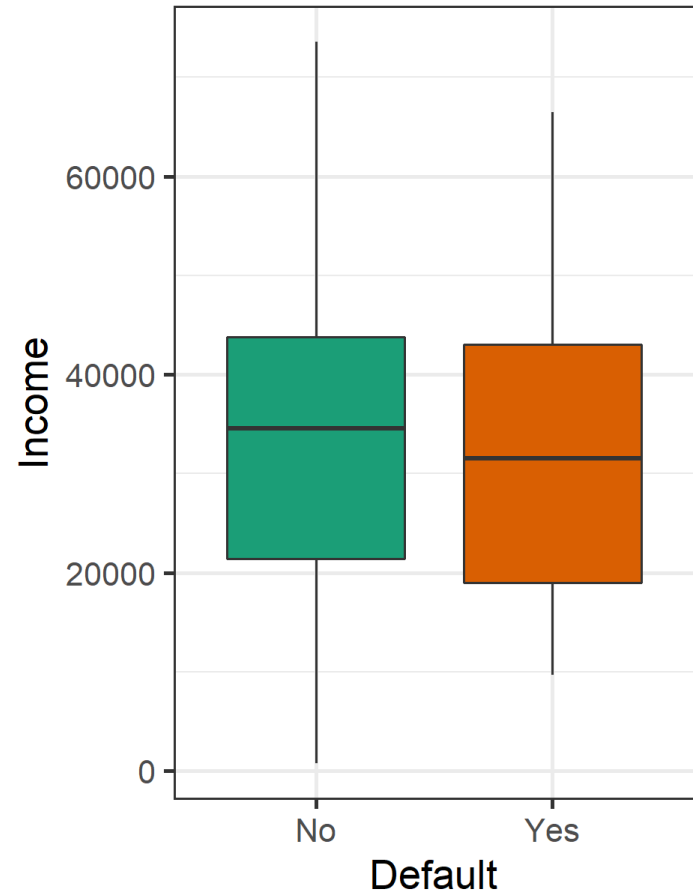
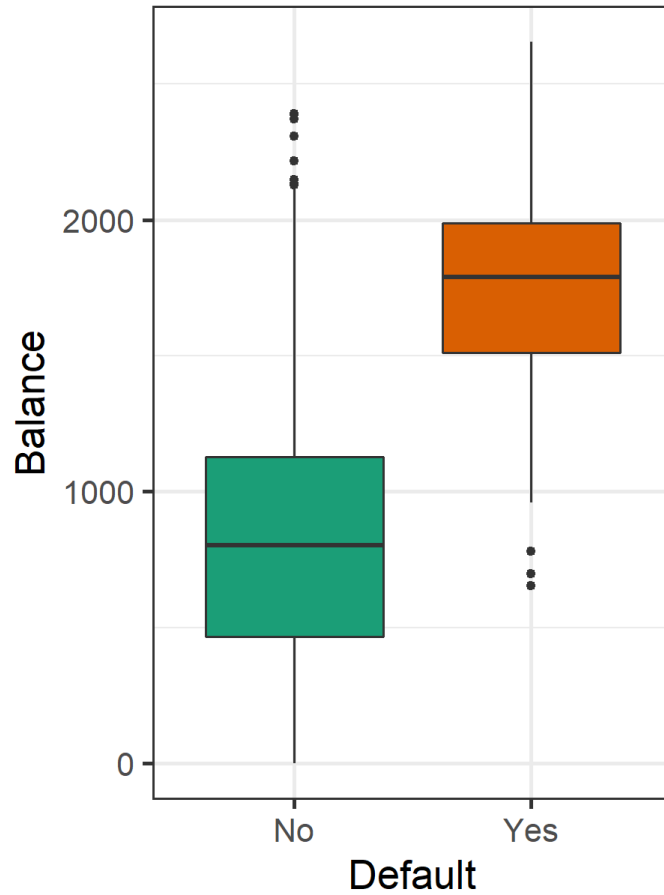
1000

Next

# Análise exploratória

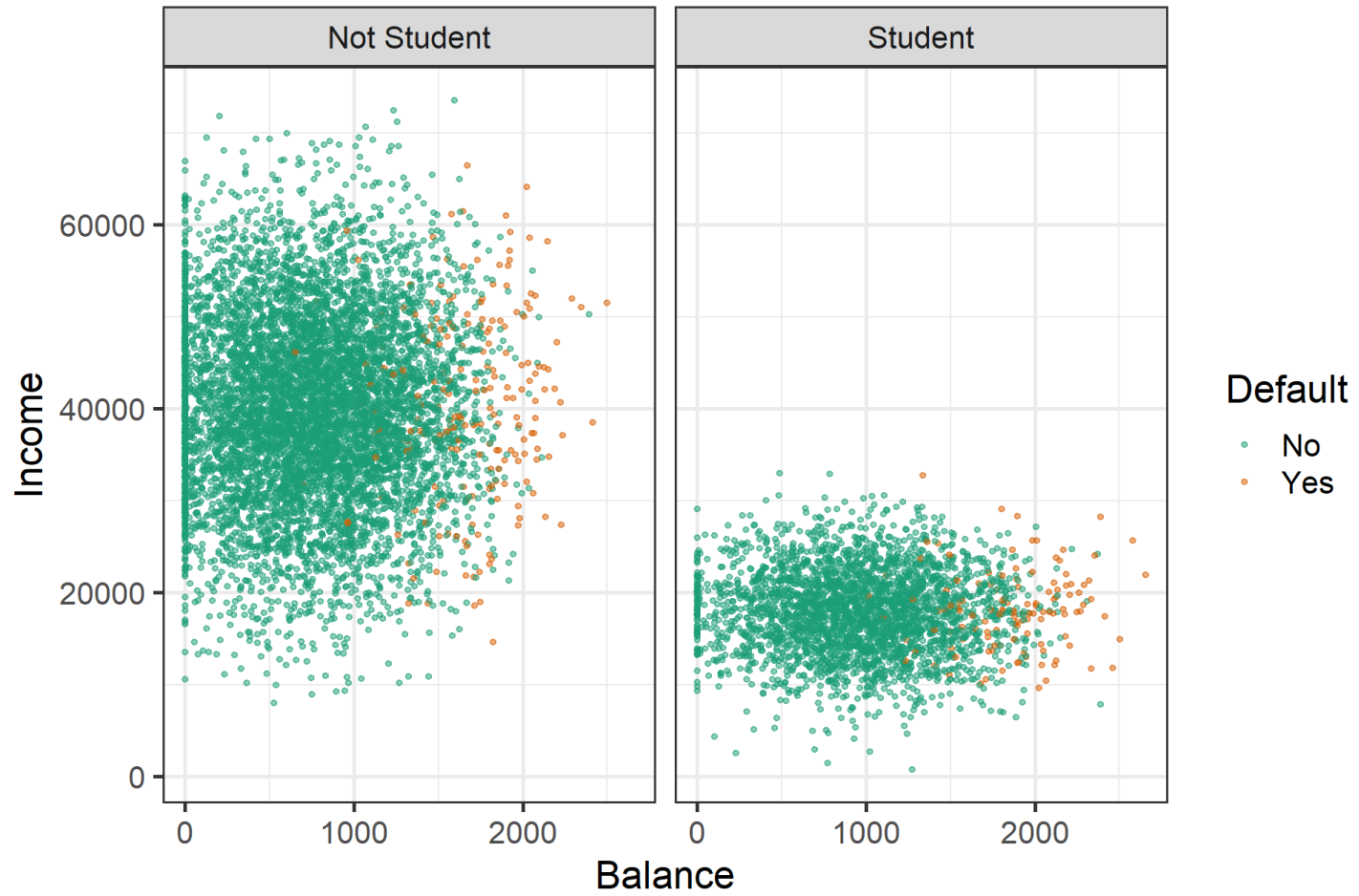


# Análise exploratória



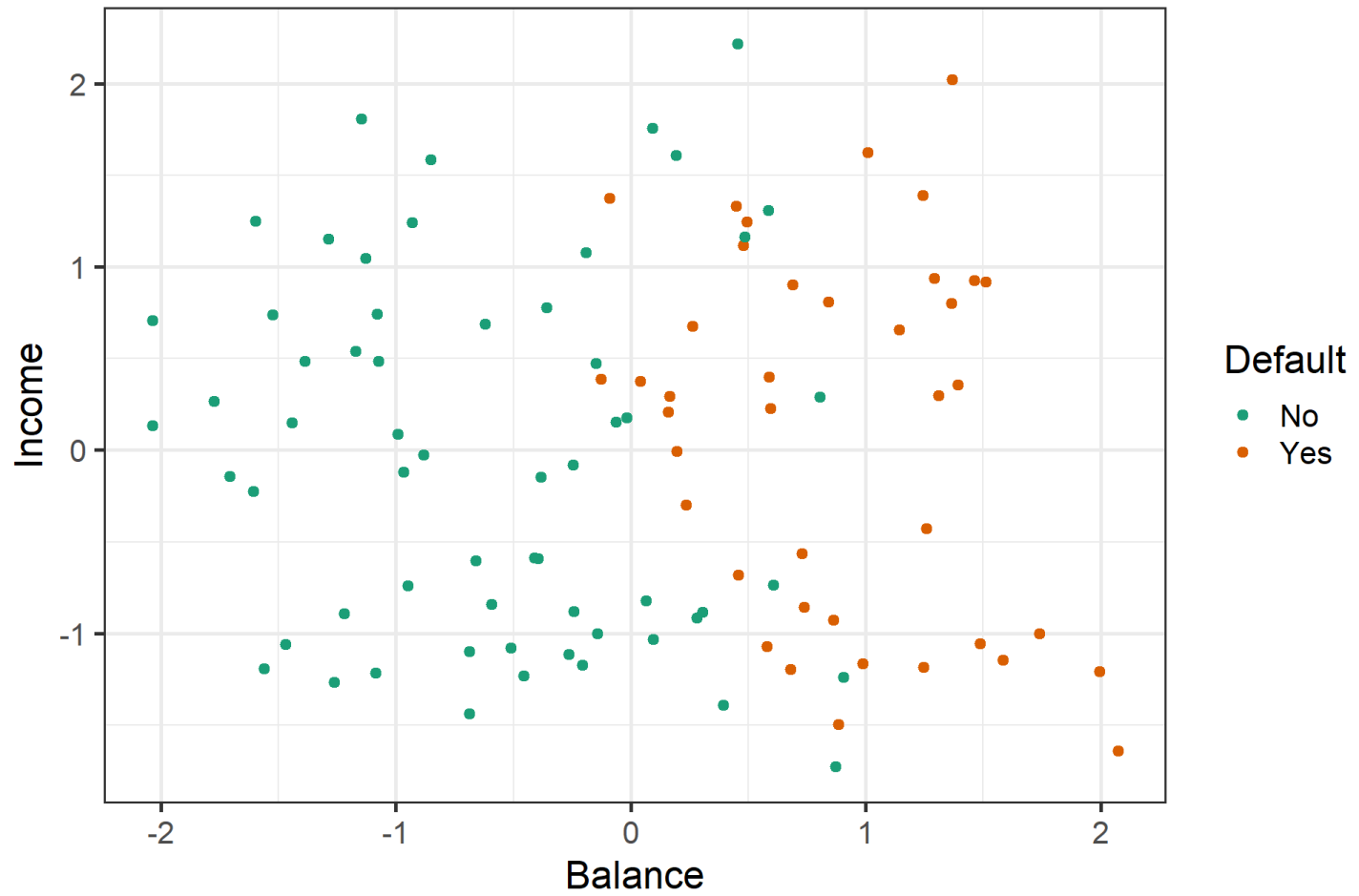


# Análise exploratória



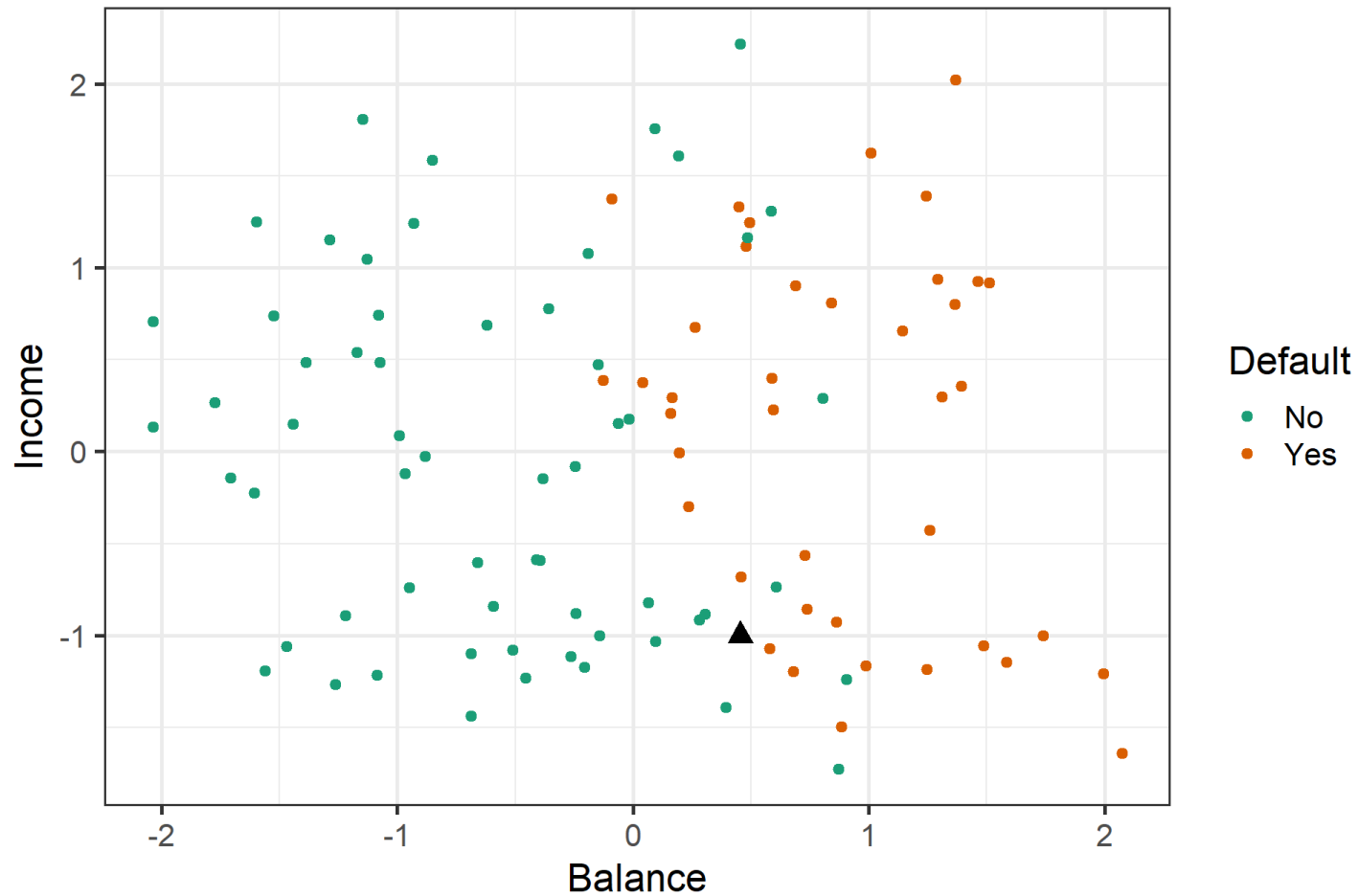
# KNN para classificação

# KNN para classificação



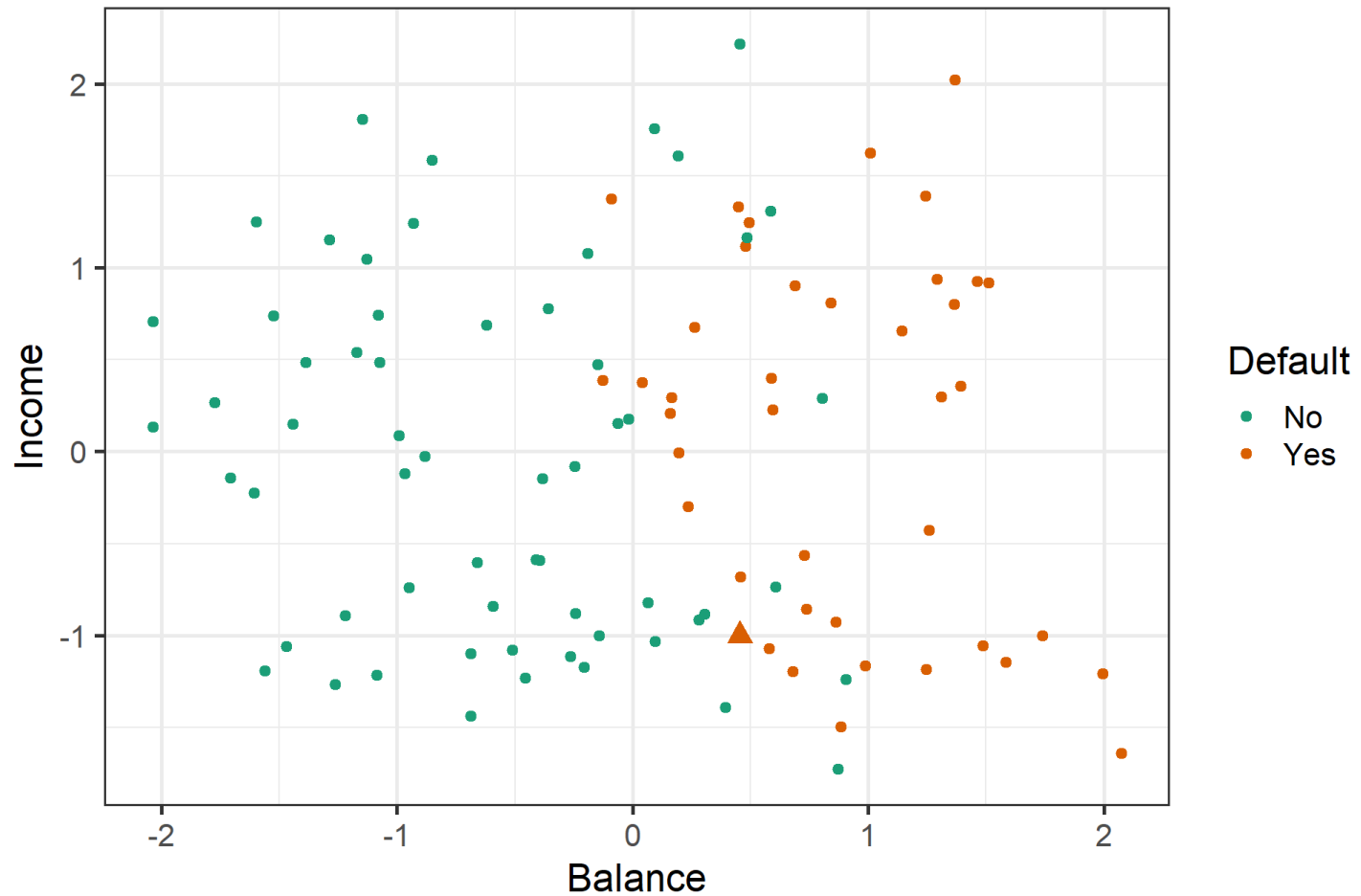
# KNN para classificação

Qual seria a previsão para o ponto preto no gráfico?



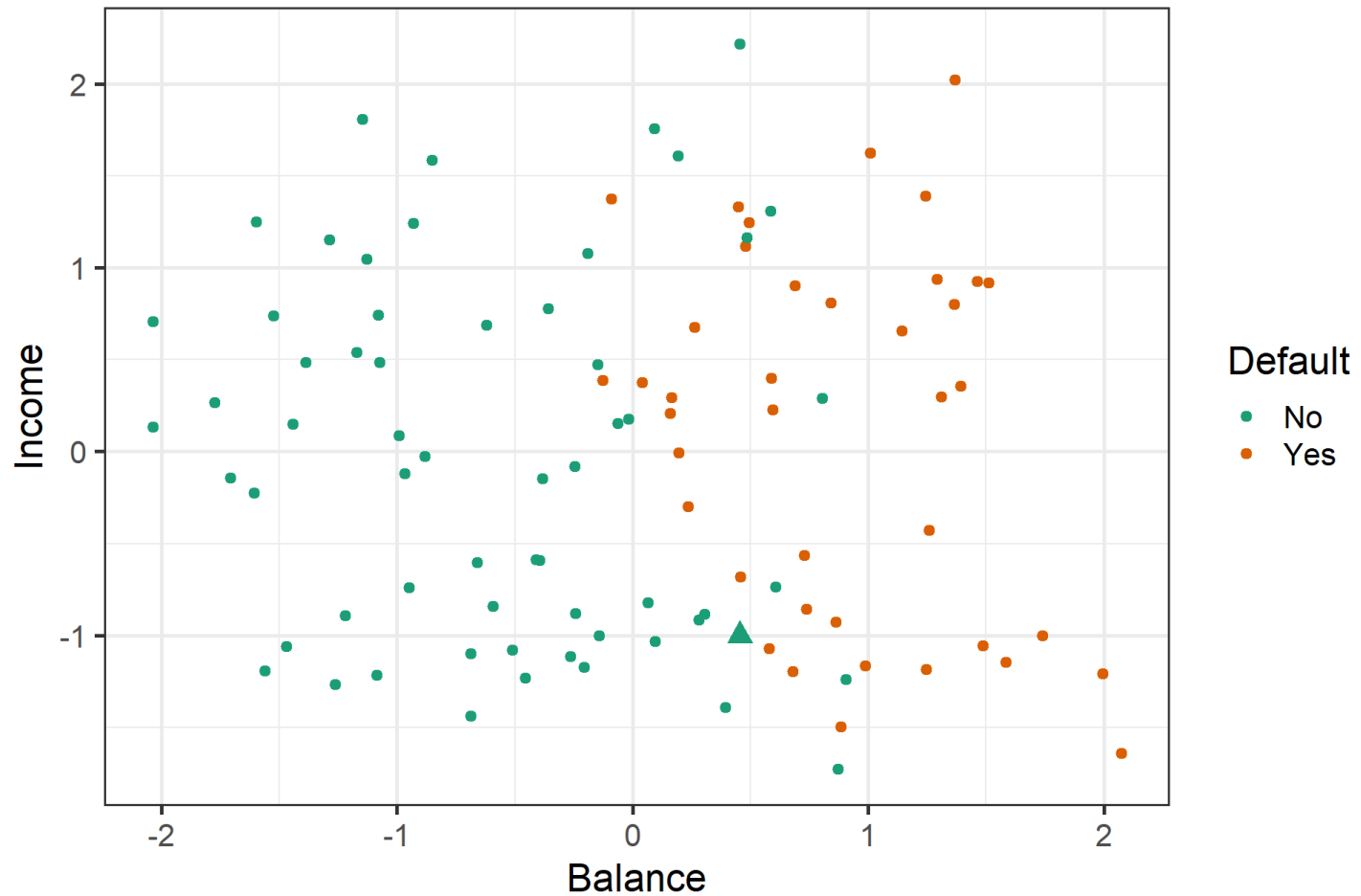
# KNN para classificação

Considerando  $k = 1$ .



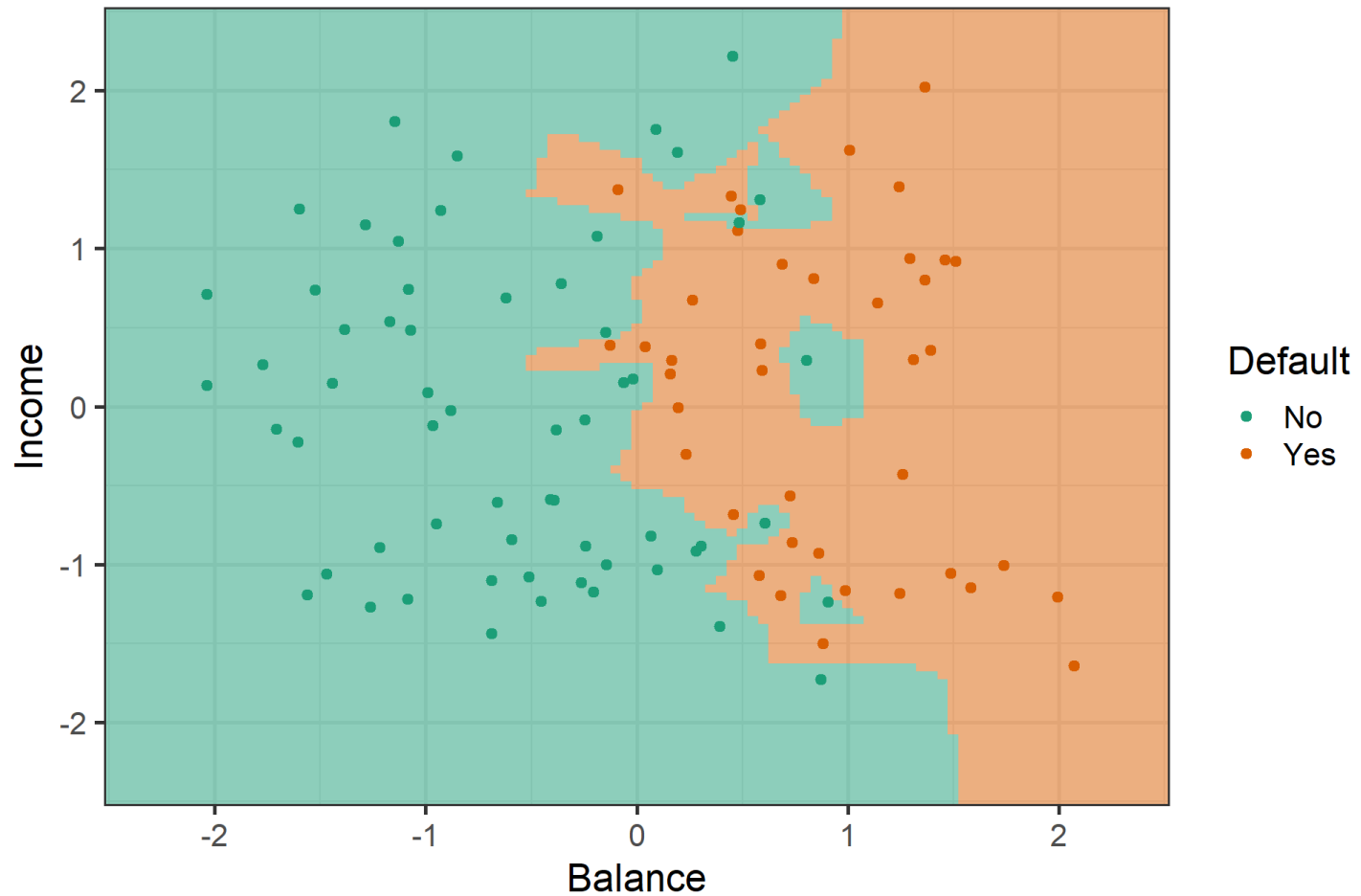
# KNN para classificação

Considerando  $k = 3$ .



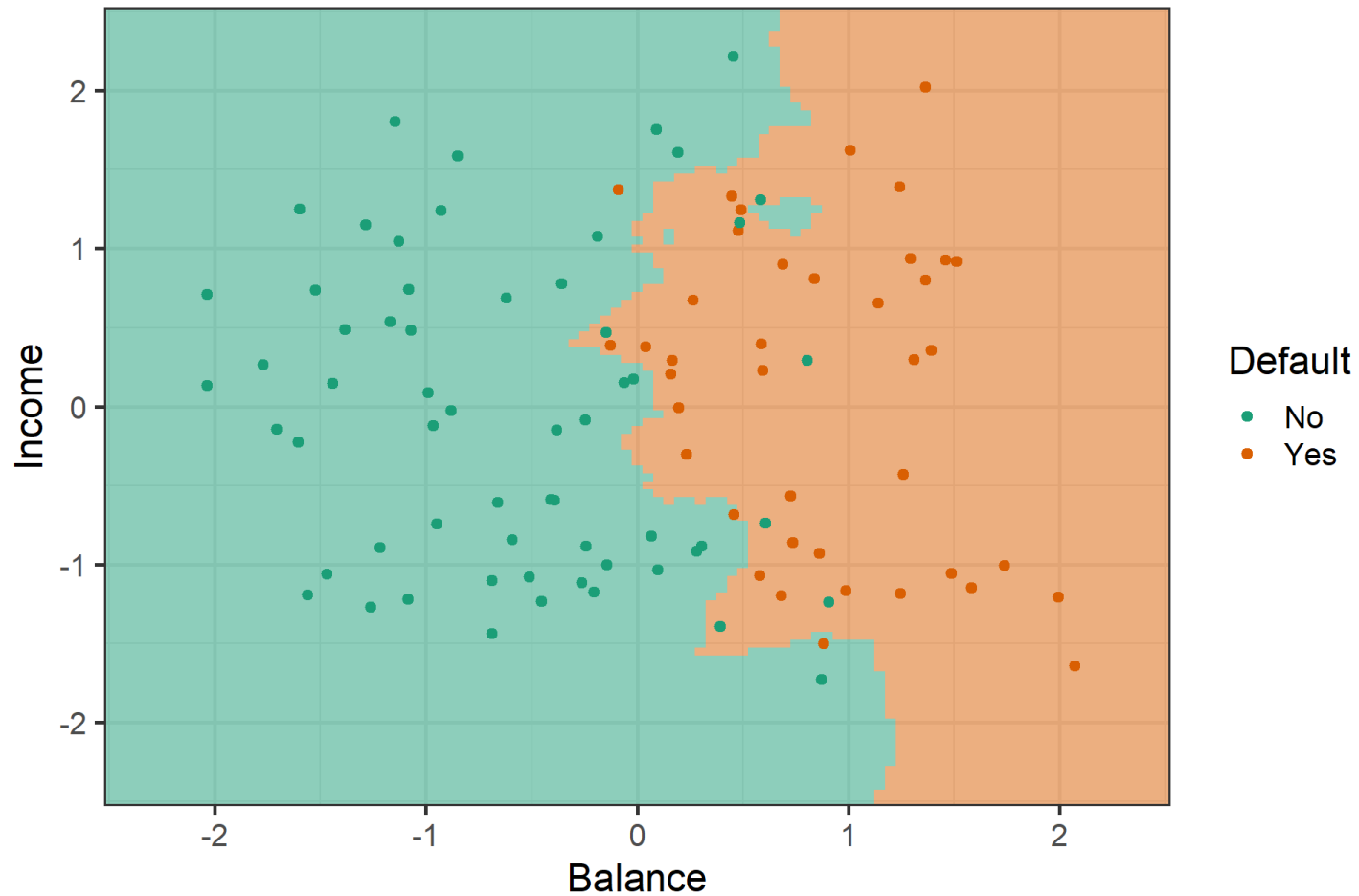
# KNN para classificação

Considerando  $k = 1$ .



# KNN para classificação

Considerando  $k = 3$ .





# Regressão logística

# Classificação

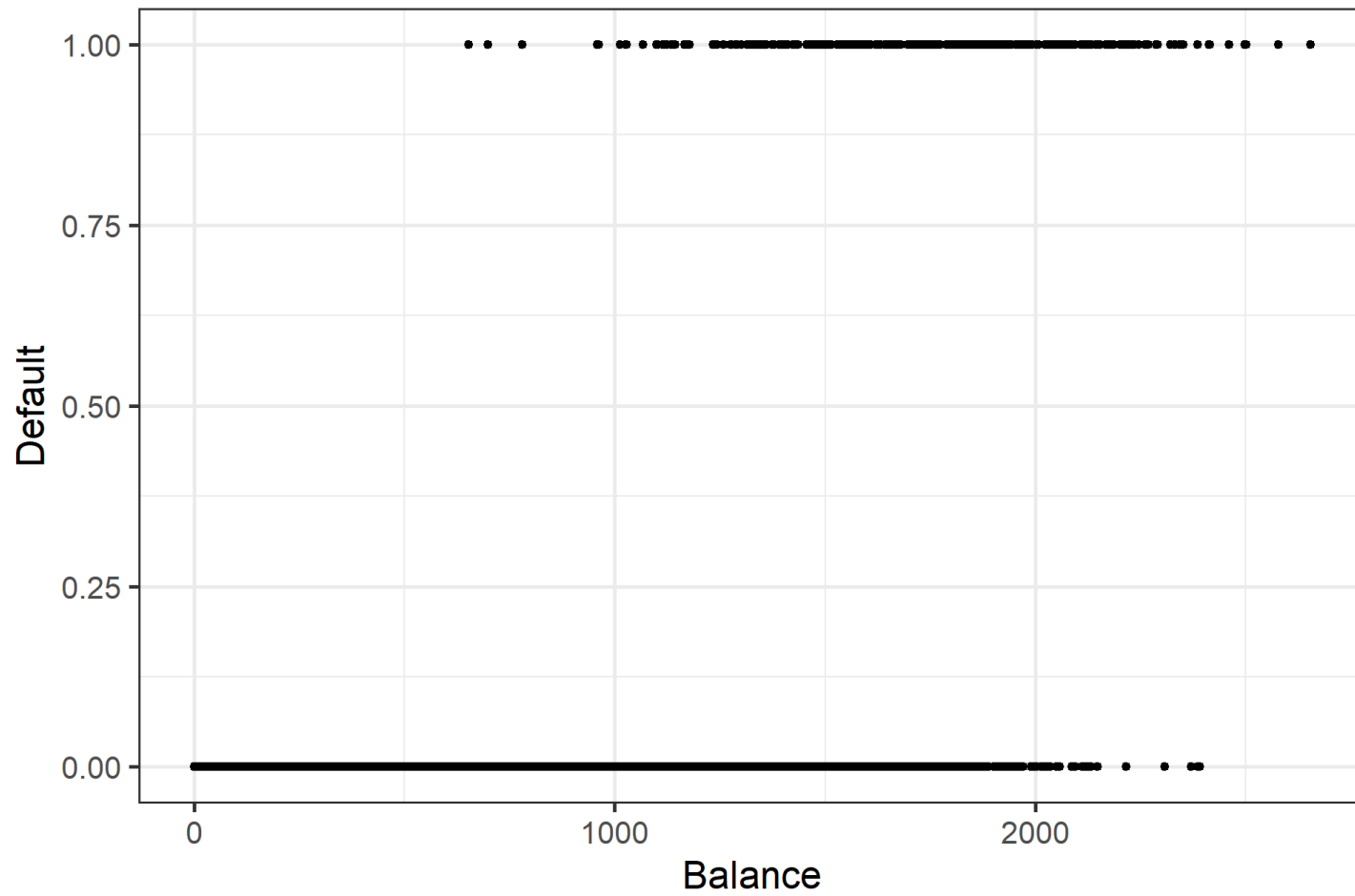
- A variável `default` assume dois possíveis valores: `Yes` e `No`.
- Ao invés de modelar diretamente a variável resposta  $Y$ , vamos modelar a *probabilidade* de  $Y$  pertencer a uma categoria em particular.
- Por exemplo, a probabilidade de `default = Yes` dado a variável `balance` pode ser escrita como

$$p(\text{balance}) = P(\text{default} = \text{Yes} | \text{balance}).$$

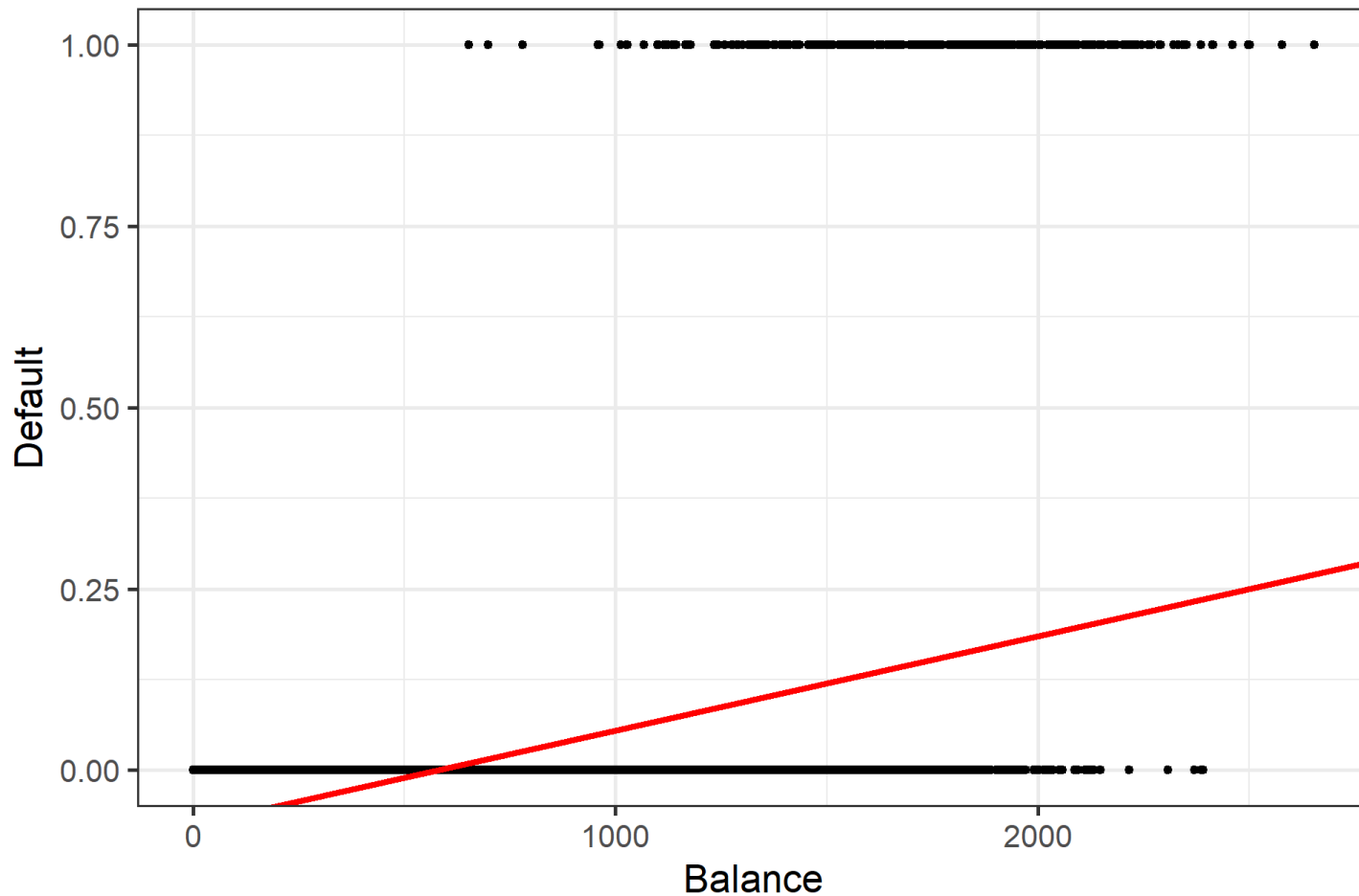
- Poderíamos, por exemplo modelar  $p(\text{balance})$  por

$$p(\text{balance}) = \beta_0 + \beta_1 \times \text{balance}.$$

# Classificação



# Porque não usar regressão linear?



Problema?

# Alternativa: modelar uma função da chance

$$\text{chance} = \frac{p(X)}{1 - p(X)}.$$

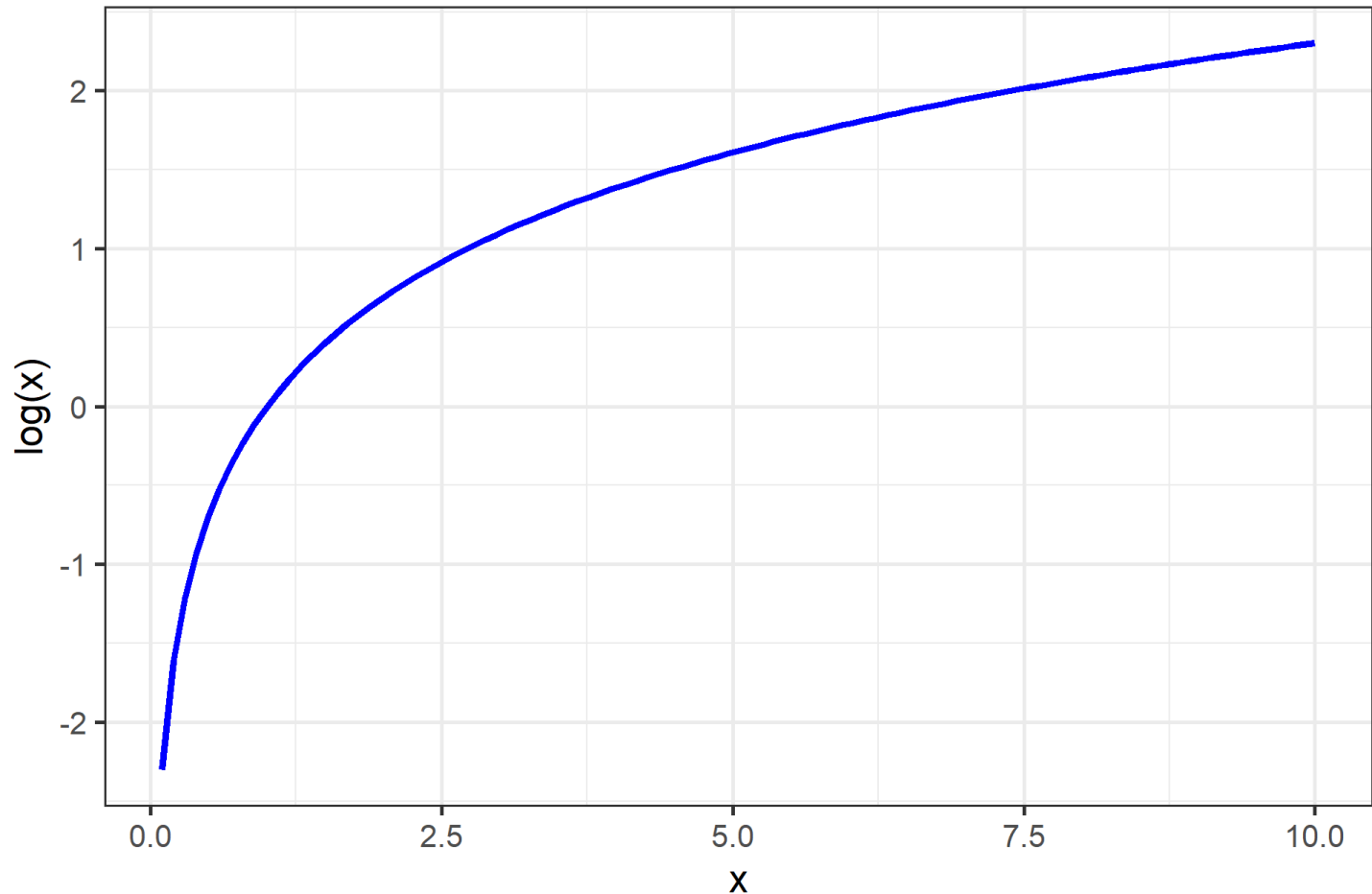
Probabilidade	Chance
0.90	90:10 ou 9
0.75	75:25 ou 3
0.50	50:50 ou 1
0.20	20:80 ou 0.25
0.10	10:90 ou 0.11
0.01	1:99 ou 0.01

# Alternativa: modelar uma função da chance

$$\text{chance} = \frac{p(X)}{1 - p(X)}.$$

Probabilidade	Chance	Log da chance
0.90	90:10 ou 9	2.197
0.75	75:25 ou 3	1.099
0.50	50:50 ou 1	0.000
0.20	20:80 ou 0.25	-1.386
0.10	10:90 ou 0.11	-2.197
0.01	1:99 ou 0.01	-4.595

## Alternativa: modelar uma função da chance



# Log da chance

$$\log(\text{chance}) = \log \left( \frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X.$$

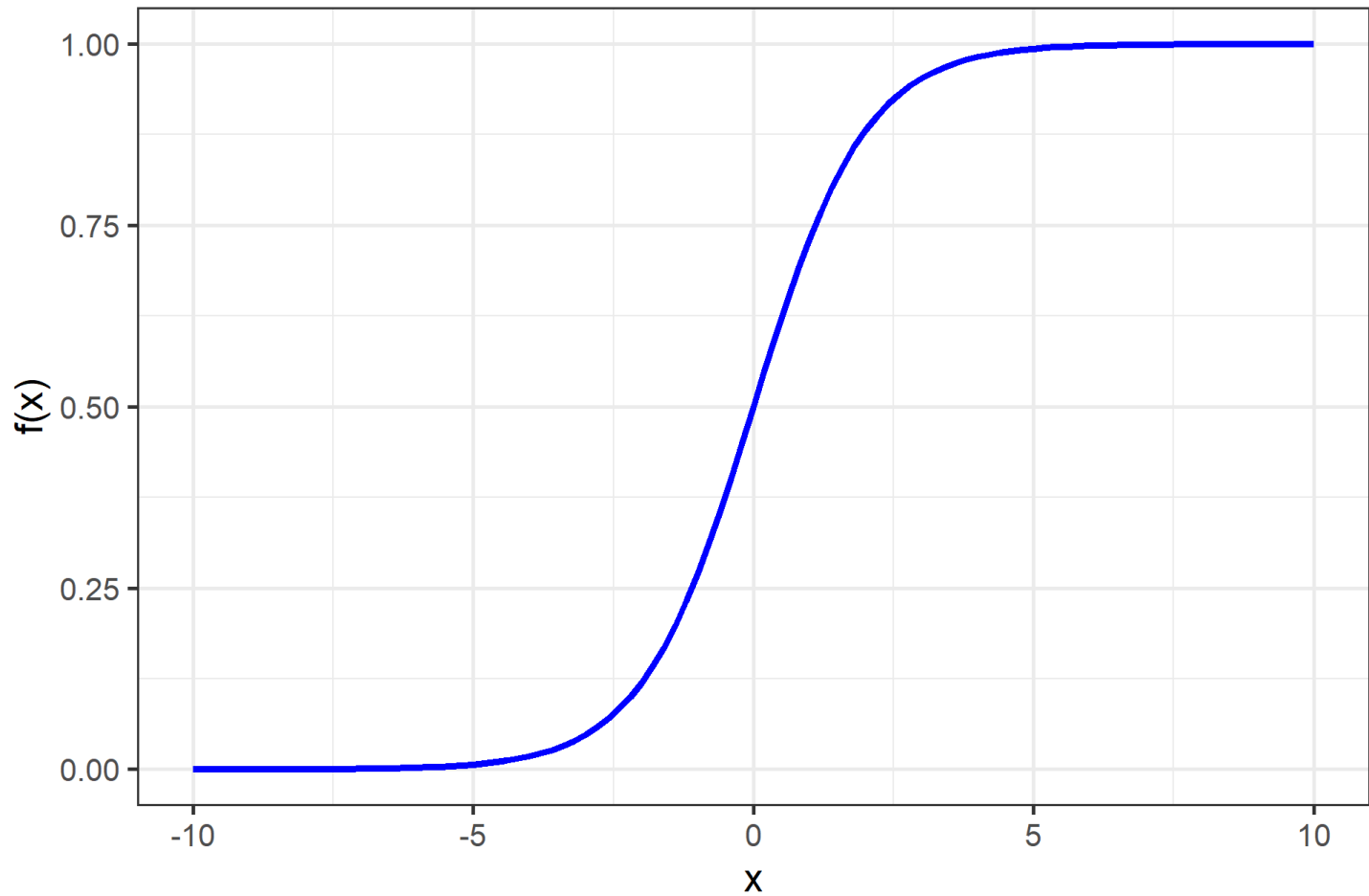
Após algumas manipulações algébricas para isolar  $p(X)$ , temos que

$$p(X) = \frac{\exp\{\beta_0 + \beta_1 X\}}{1 + \exp\{\beta_0 + \beta_1 X\}} = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 X)\}}.$$



# Logística

$$f(x) = \frac{\exp\{x\}}{1 + \exp\{x\}}$$



# Logística

$$\log\left(\frac{p(x)}{1-p(x)}\right) = -10.6513 + 0.0055x.$$

# Logística

$$\log\left(\frac{p(x)}{1-p(x)}\right) = -10.7495 + 0.0057 \times \textit{balance} - 0.7149 \times \textit{student}.$$

# Como obter as estimativas para $\beta_j$ ?

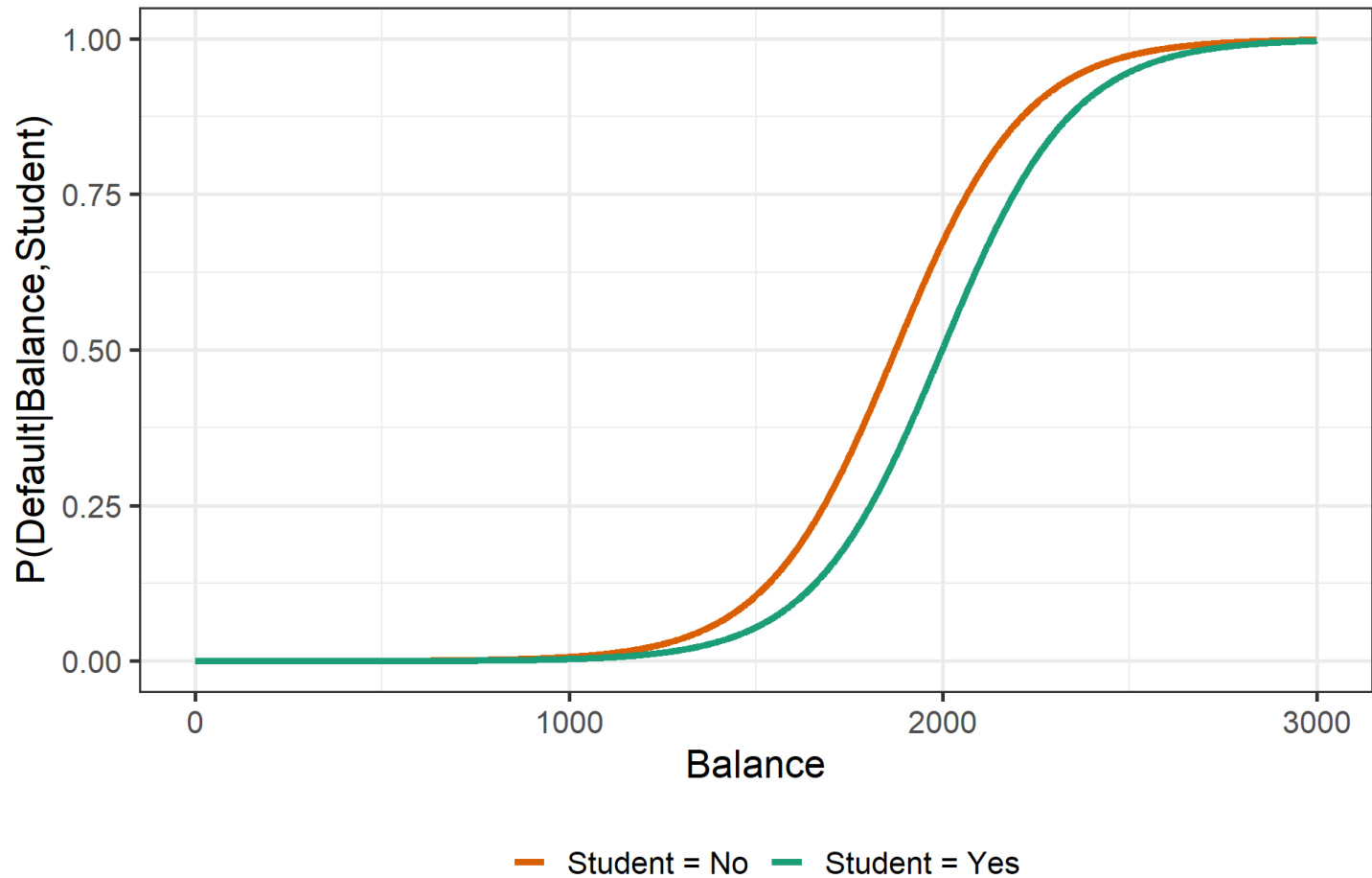
Com a função de verossimilhança.

$$\begin{aligned} L_{\mathbf{x},y}(\theta) &= P(Y_1 = y_1, \dots, Y_n = y_n | \mathbf{x}_1, \dots, \mathbf{x}_n, \beta) \\ &= \prod_{i=1}^n P(Y_i = y_i | \mathbf{x}_i, \theta) \\ &= \prod_{i=1}^n p(\mathbf{x}_i)^{y_i} [1 - p(\mathbf{x}_i)]^{1-y_i} \\ &= \prod_{i=1}^n \left( \frac{\exp\{\beta_0 + \beta_1 x_{1,i} + \dots + \beta_1 x_{p,i}\}}{1 + \exp\{\beta_0 + \beta_1 x_{1,i} + \dots + \beta_1 x_{p,i}\}} \right)^{y_i} \\ &\quad \times \left( 1 - \frac{\exp\{\beta_0 + \beta_1 x_{1,i} + \dots + \beta_1 x_{p,i}\}}{1 + \exp\{\beta_0 + \beta_1 x_{1,i} + \dots + \beta_1 x_{p,i}\}} \right)^{1-y_i}. \end{aligned}$$

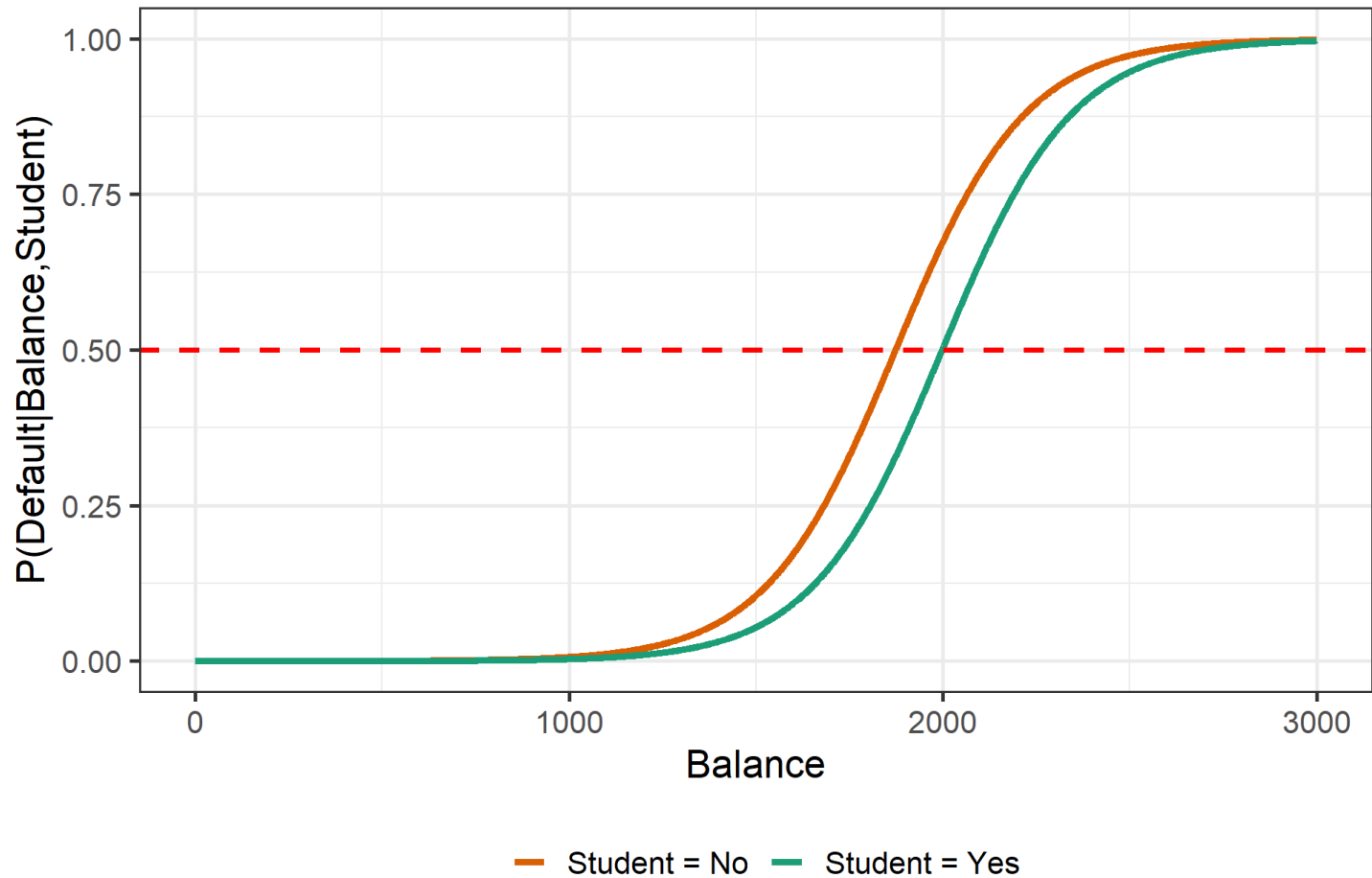
# Modelo estimado

```
##
## Call:
## glm(formula = default ~ balance + student, family = binomial,
##      data = Default)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4578  -0.1422  -0.0559  -0.0203   3.7435
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.075e+01  3.692e-01 -29.116  < 2e-16 ***
## balance      5.738e-03  2.318e-04  24.750  < 2e-16 ***
## studentYes  -7.149e-01  1.475e-01  -4.846  1.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.7  on 9997  degrees of freedom
## AIC: 1577.7
##
## Number of Fisher Scoring iterations: 8
```

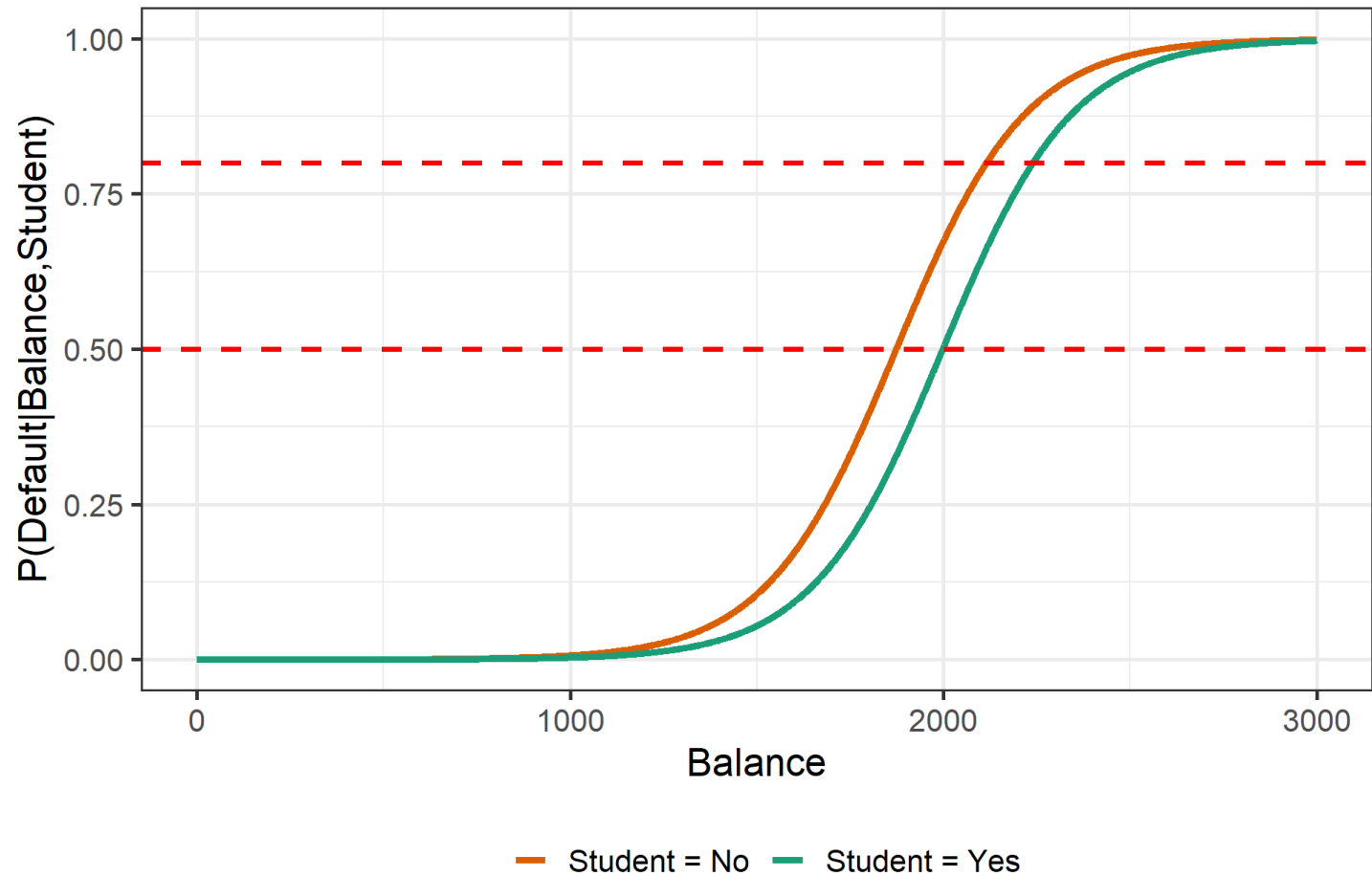
# Como classificar?



# Como classificar?



# Como classificar?





# Matriz de confusão

Para corte = 0.5.

##		Observado	
##	Predito	No	Yes
##	No	9628	228
##	Yes	39	105

Para corte = 0.8.

##		Observado	
##	Predito	No	Yes
##	No	9663	303
##	Yes	4	30

# Matriz de confusão

Classificado	Observado	
	No	Yes
No	Verdadeiro negativo	Falso negativo
Yes	Falso positivo	Verdadeiro

# Métricas

Classificado	Observado	
	No	Yes
No	a	b
Yes	c	d

**Erro de classificação total:**  $\frac{b+c}{n} = 1 - \frac{a+d}{n}$ ;

**Verdadeiro positivo (sensibilidade ou recall):**  $\frac{d}{b+d}$ ;

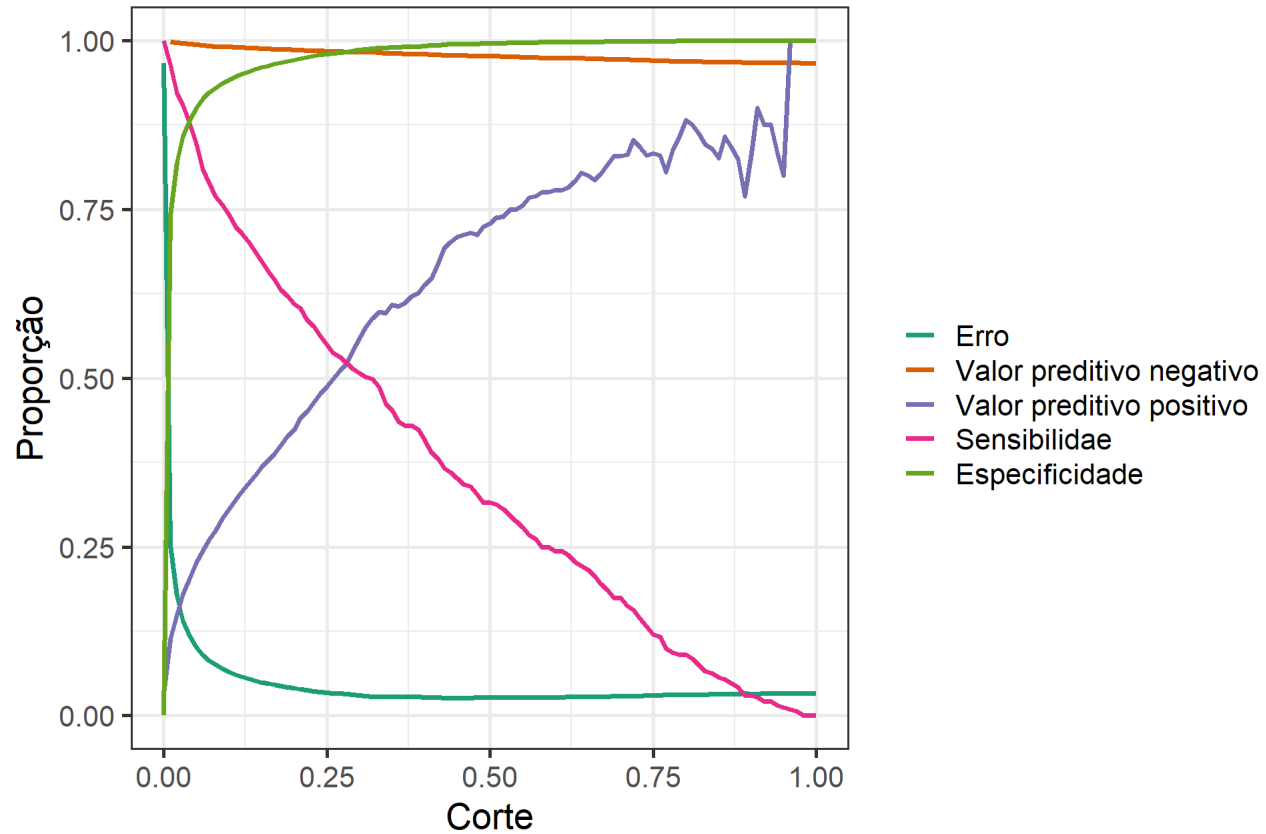
**Verdadeiro negativo (especificidade):**  $\frac{a}{a+c}$ ;

**Valor preditivo positivo (precision):**  $\frac{d}{c+d}$ ;

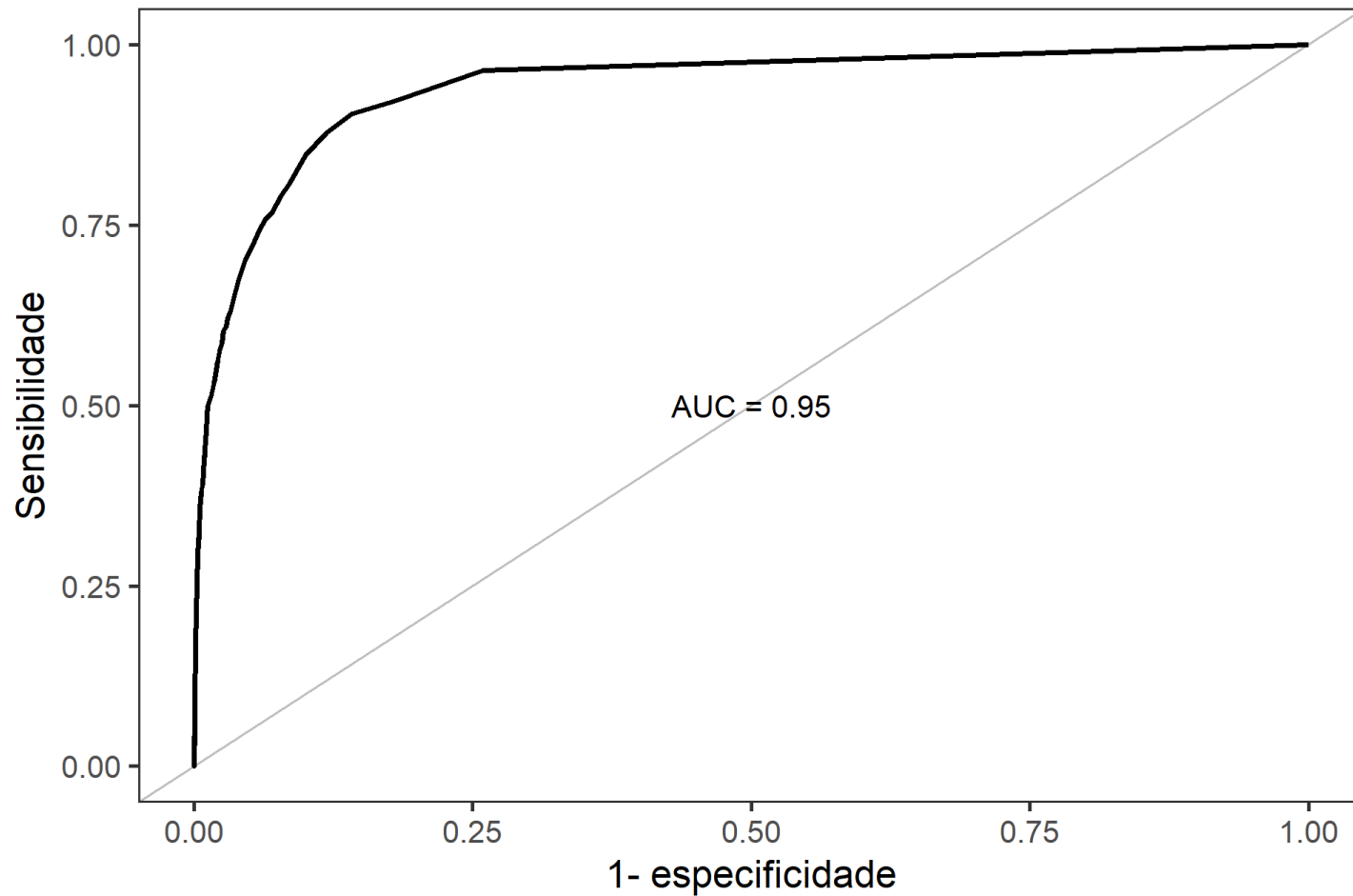
**Valor preditivo negativo:**  $\frac{a}{a+b}$ ;

**F-score:**  $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$ .

# Medidas



# Curva ROC



# Adicionando fator de perda/ganho na classificação

Considere os seguintes ganhos/perdas a depender da classificação feita por um dado modelo.

Classificado	Observado	
	No	Yes
No	10	-5
Yes	-20	100

Em um conjunto com 100 observações, obteve-se o seguinte cenário.

Classificado	Observado	
	No	Yes
No	30	20
Yes	10	40

Então, o lucro esperado será

$$\begin{aligned}\text{Lucroesperado} &= 10 \times \frac{30}{100} + (-5) \times \frac{20}{100} + (-20) \times \frac{10}{100} + 100 \times \frac{40}{100} \\ &= 40.\end{aligned}$$

# Dados desbalanceados <sup>1</sup>

Há algumas alternativas para situações em que as classes estão desbalanceadas:

- Ajustar o modelo para maximizar a acurácia da classe minoritária;
- Escolher o corte para classificação com base na curva ROC;
- Poderar os dados com pesos maiores para as classes minoritárias;
- *Down-sampling*: amostra dados da classe majoritária para que tenha a mesma proporção da classe minoritária;
- *Up-sampling*: é feito um processo de reamostragem com reposição do grupo minoritário até tenha aproximadamente o mesmo número de observações que o grupo majoritário.

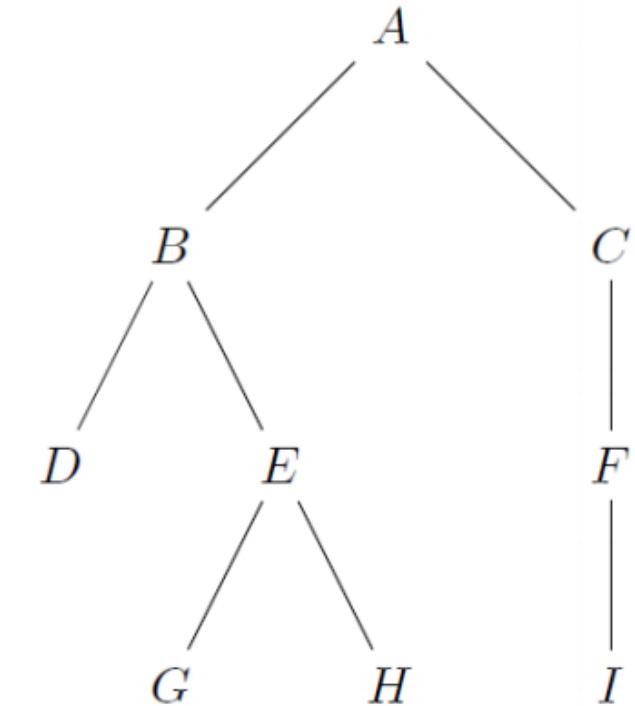
[1] Fonte: livro *Applied predictive modeling*

# Árvores de classificação



# Terminologia

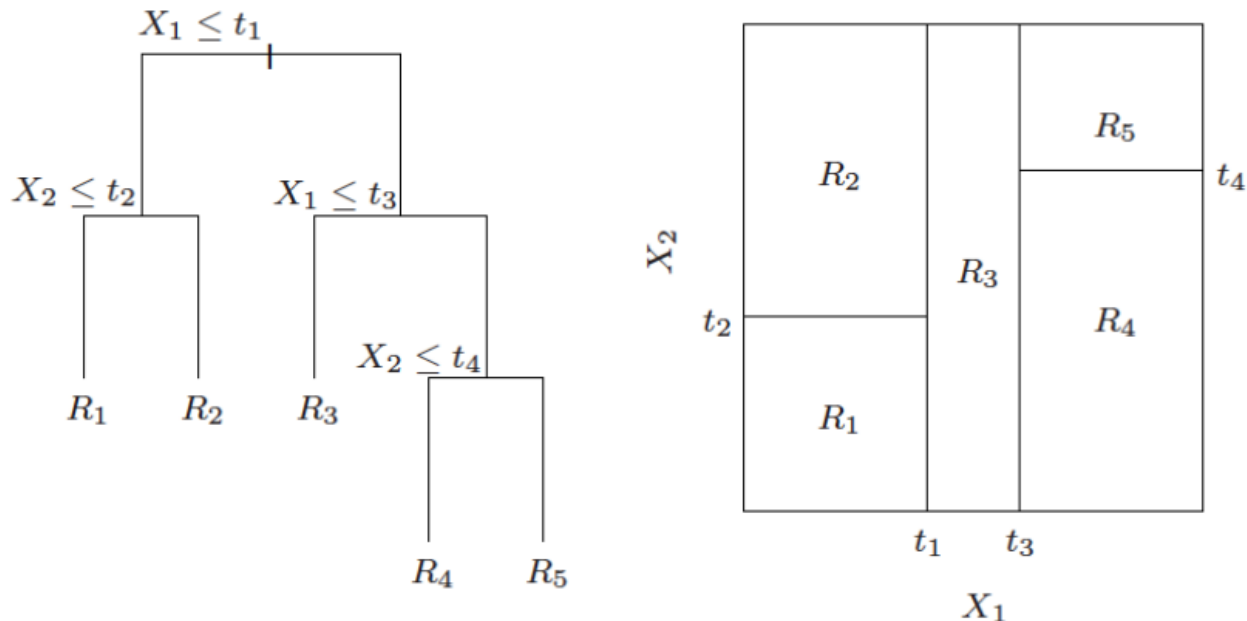
Na *árvore binária* abaixo, o nó *A* é a *raiz* e os nós *D*, *G*, *H* e *I* são as *folhas* (ou *nós terminais*).



A árvore tem quatro níveis de *altura*. A *profundidade* do nó interno *D* é igual a dois.

# Regiões determinadas pela árvore de classificação

Na figura abaixo, temos uma árvore de classificação  $T$  e a partição do espaço das preditoras nas regiões correspondentes.



Cada nó não terminal de  $T$  define uma divisão (*split*) em uma das preditoras. Cada folha de  $T$  corresponde a uma região retangular  $R_j$ .

# Como classificar?

- Suponha que a árvore  $T$  do slide anterior nos foi dada, construída a partir de dados de treinamento

$$(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2 \times \{0, 1\}.$$

- Tendo em mãos um novo  $x \in \mathbb{R}^2$ , determinamos a qual região  $R_j$  este dado  $x$  pertence.
- Note que não precisamos examinar todas as regiões: basta descer a árvore a partir da raiz para saber a qual região  $x$  pertence.
- Do ponto de vista computacional, é uma grande vantagem!
- Uma vez determinada a região  $R_j$  a qual  $x$  pertence, classificamos este dado como sendo da classe mais frequente entre os dados de treinamento pertencentes à mesma região  $R_j$  (voto da maioria).

# De maneira formal

- Dados de treinamento:  $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2 \times \{0, 1\}$ , para  $i = 1, \dots, n$ . Considere que os dados podem ser categorizados em  $m$  classes.
- A árvore de classificação  $T$  define as regiões retangulares  $R_1, \dots, R_m$  que particionam o espaço das preditoras  $\mathbb{R}^p$ .
- Seja  $n_j = \sum_{i=1}^n \mathbb{I}_{R_j}(x_i)$  o número de observações dos dados de treinamento pertencentes à região  $R_j$ , para  $j = 1, \dots, m$ .
- A fração de elementos da classe  $k$  na região  $R_j$  é igual a

$$\hat{p}_k(R_j) = \frac{1}{n_j} \sum_{\{i: x_i \in R_j\}} \mathbb{I}_{\{k\}}(y_i),$$

para  $k = 1, \dots, c$  e  $\mathbb{I}_{\{k\}}(y_i)$  sendo a indicadora de que  $y_i$  pertence à classe  $k$ .

- A classe predita para a região  $R_j$  é  $c_j = \arg \max_k \hat{p}_k(R_j)$ , que é a proporção de observações de treinamento na região  $R_j$  que são da classe predominante.
- Finalmente, o classificador fica escrito como

# Como construir uma árvore

- Para construir cada divisão da árvore, precisamos escolher uma das preditoras e o ponto de separação.
- Como escolher cada uma das divisões (*splits*)?
- Qual altura a árvore deve ter?
- Qual algoritmo utilizar?

# Algoritmo CART <sup>1</sup>

- CART: Classification and Regression Trees.
- O algoritmo CART começa na raiz da árvore e efetua uma divisão, criando dois nós no próximo nível da árvore.
- Depois disso, descemos para o primeiro nível da árvore e repetimos o procedimento para os dois nós que foram criados.
- Continuamos da mesma maneira nos níveis seguintes.
- Em cada etapa, escolhemos a divisão que produz a maior queda no erro de classificação.
- O algoritmo CART cresce uma árvore alta e poda alguns dos seus ramos no final do processo.

[1] Fonte: livro *Classification and Regression Trees*, Breiman et al., 1984.

# Algoritmo CART

- Formalmente, o algoritmo CART começa na raiz da árvore e define as regiões disjuntas

$$R_1 = \{X \in \mathbb{R}^p : X_j \leq t\} \quad \text{e} \quad R_2 = \{X \in \mathbb{R}^p : X_j > t\}.$$

- Utilizando os dados de treinamento, fazemos a divisão escolhendo  $\hat{j}$  e  $\hat{t}$  tais que

$$(\hat{j}, \hat{t}) = \arg \min_{(j,t)} \{(1 - \hat{p}_{c_1}(R_1)) + (1 - \hat{p}_{c_2}(R_2))\},$$

em que

$$c_1 = \arg \max_{\{k=1,\dots,c\}} \hat{p}_k(R_1)$$

é a classe dominante no retângulo  $R_1$  e

$$c_2 = \arg \max_{\{k=1,\dots,c\}} \hat{p}_k(R_2)$$

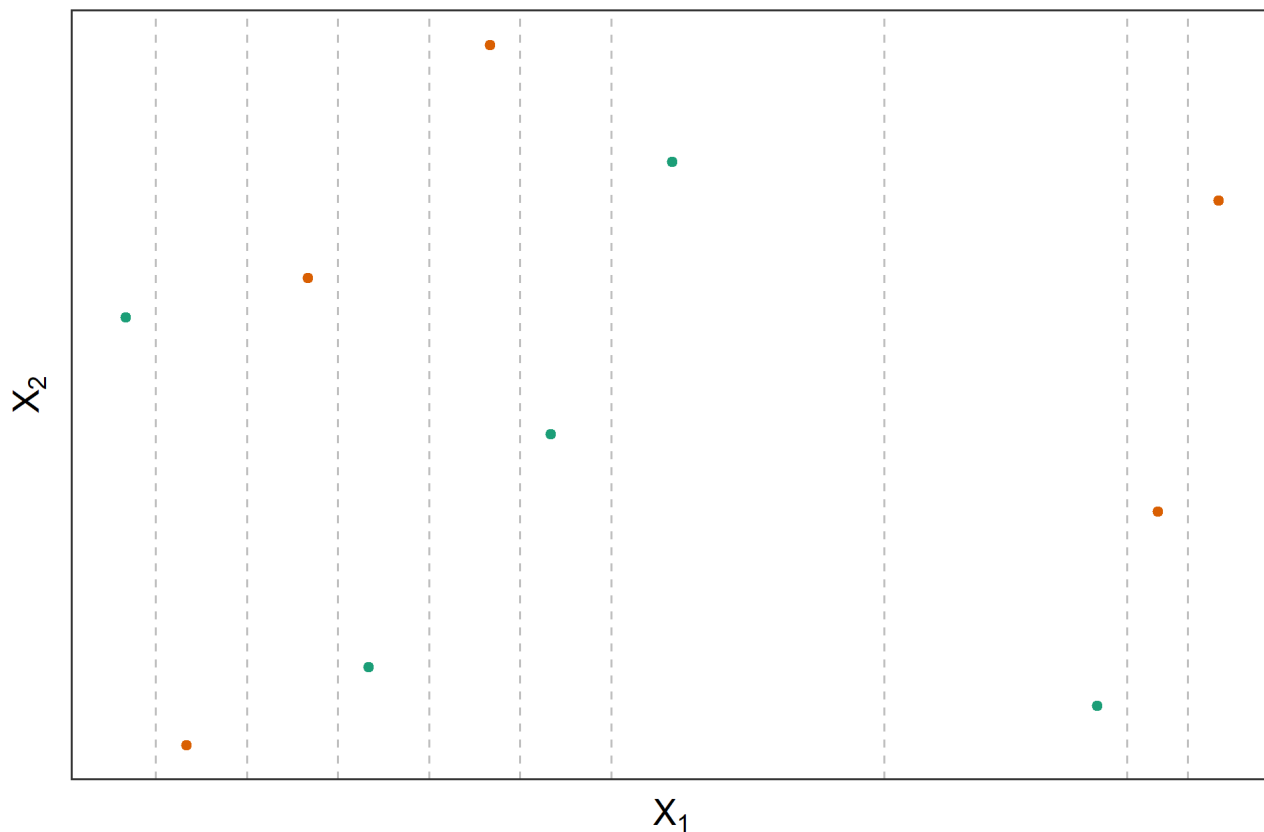
é a classe dominante no retângulo  $R_2$ .

- Note que  $1 - \hat{p}_{c_1}(R_1)$  é a fração dos dados de treinamento que é classificada incorretamente no retângulo  $R_1$  e  $1 - \hat{p}_{c_2}(R_2)$  é o análogo no retângulo  $R_2$ .

# Algoritmo CART

Para encontrar o ponto  $t$  de divisão de uma região retangular  $R_m$ , precisamos considerar apenas  $n_m - 1$  divisões da variável preditora  $X_j$  que estivermos examinando.

Na figura abaixo, estamos examinando a primeira divisão na variável  $X_1$

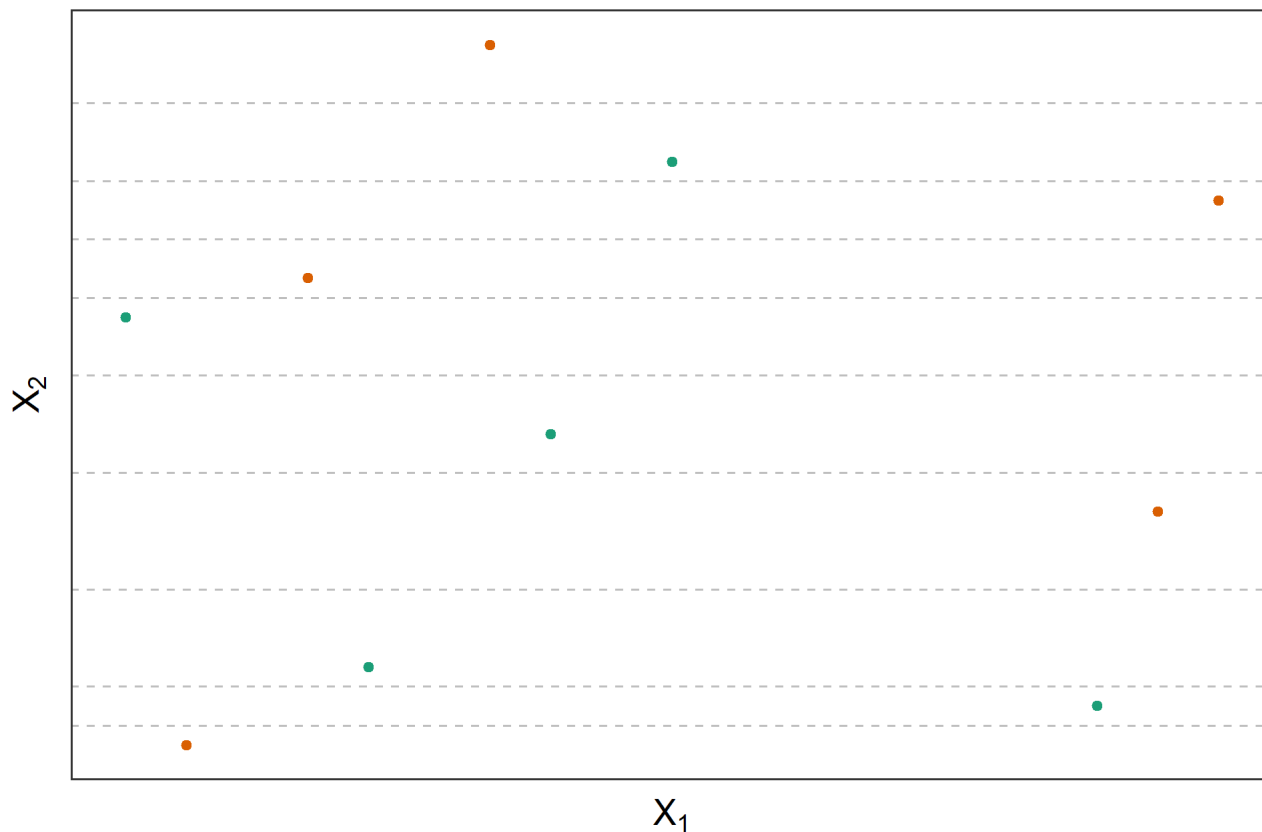




# Algoritmo CART

Para encontrar o ponto  $t$  de divisão de uma região retangular  $R_m$ , precisamos considerar apenas  $n_m - 1$  divisões da variável preditora  $X_j$  que estivermos examinando.

Na figura abaixo, estamos examinando a primeira divisão na variável  $X_2$



# Algoritmo CART

Procedemos de maneira análoga para os novos nós criados, até que seja satisfeito algum critério de parada; por exemplo, quando tivermos apenas dados de treinamento de uma certa classe na nova região gerada, ou um número mínimo de elementos em cada região.

Este procedimento gera uma árvore  $T_0$  que será podada: algumas de suas folhas serão colapsadas aos seus nós pais.

Para uma árvore de classificação  $T$ , denote por  $|T|$  o número de suas folhas e, para  $\alpha \geq 0$ , defina

$$C_\alpha(T) = \sum_{j=1}^{|T|} (1 - \hat{p}_{c_j}(R_j)) + \alpha|T|.$$

O algoritmo CART escolhe a árvore  $T$  que minimiza  $C_\alpha(T)$ , escolhendo  $\alpha$  por validação cruzada.

Note que há uma forma de regularização contida na definição de  $C_\alpha$ , uma vez que estamos penalizando árvores com muitas folhas.

# Prática R

# Árvores de regressão

Lembre-se que em um problema de regressão, a variável resposta é quantitativa,  $Y \in \mathbb{R}$ .

O algoritmo CART constrói a árvore de regressão de maneira análoga ao caso de classificação.

A principal diferença é que para definir as divisões utilizamos uma perda quadrática ao invés do erro de classificação

$$(\hat{j}, \hat{t}) = \arg \min_{(j,t)} \left\{ \sum_{\{i: x_i \in R_1\}} (y_i - \hat{y}_{R_1})^2 + \sum_{\{i: x_i \in R_2\}} (y_i - \hat{y}_{R_2})^2 \right\},$$

em que  $\hat{y}_{R_1}$  e  $\hat{y}_{R_2}$  são as médias das respostas dos dados de treinamento que pertencem às regiões  $R_1$  e  $R_2$ , respectivamente.

# Árvores de regressão

Para cada região  $R_j$  correspondente a um nó terminal da árvore de regressão obtida, o CART associa uma constante  $c_j$  que é a média das respostas dos dados de treinamento que pertencem à região  $R_j$ ,

$$c_j = \frac{1}{n_j} \sum_{\{i: x_i \in R_j\}} y_j.$$

Então, a estimativa CART para a função de regressão é

$$\hat{f}(x) = \sum_j c_j \mathbb{I}_{R_j}(x).$$

# Vantagens e desvantagens

## Aspectos Positivos

- Fácil de explicar (muito mais que regressão linear);
- Podem ser apresentadas graficamente e facilmente interpretadas por pessoas que não são especialistas no assunto;
- Tratam facilmente preditores qualitativos, sem a necessidade da criação de variáveis indicadoras / *dummies*;
- Não é sensível à escala como outros métodos.

## Aspectos Negativos

- Uma pequena alteração nos dados pode causar uma grande alteração na árvore estimada (variância alta);
- Previsões baseadas em regiões retangulares;
- Não apresentam desempenho preditivo tão bom quanto outros métodos.

# Resumindo...

- Definição de um problema de classificação.
- kNN para classificação.
- Regressão logística.
- Árvore de classificação e regressão.
- Métricas para avaliar a performance de um modelo de classificação.

**Obrigado!**

**`magnotfs@insper.edu.br`**